REMARKS

Claims 1-8, 10-15, 17-23, and 27 are currently pending in the subject application and are presently under consideration. Claims 1, 14, 15, 23, and 27 have been amended as shown on pp. 2 and 4-7 of the Reply. An interview was conducted on April 29, 2008. While the discussion generally related to all the claims, the focus of the discussion focused on independent claims 1 and 27. Amendments disclosed herein were presented to the Examiner. The interview was conducted with Ronald Krosky (Reg. No. 58,564), Asmita Chande and Examiner Chankong. The main focus of the interview was on deficiencies of the 101, 112 and 103 rejections. No agreements were reached. Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-8, 10-14 and 27 Under 35 U.S.C. §101

Claims 1-8, 10-14 and 27 stand rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. This rejection should be withdrawn for at least the following reasons. The Federal Circuit has clearly established in *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005) and *AT&T Corp. v. Excel Communications, Inc*., 172 F.3d 1352, 1358. (Fed.Cir. 1999) that inventions such as that claimed by applicant are statutory.

> This court must also decide whether software code made in the United States and exported abroad is a "component of a patented invention" under 271(f)... Section 271(f) refers to "components of a patented invention."... Title 35, section 101, explains that an invention includes "any new and useful process, machine, manufacture or composition of matter."... Without question, ***software code alone qualifies as an invention eligible for patenting under these categories***, at least as processes. *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005). (Emphasis added).

The Federal Circuit in *Eolas Techs., Inc. v. Microsoft Corp.* clearly established that software code alone is statutory subject matter. Independent claims 1 and 27, as amended, relate to a system *implemented on a computer*. A computer implemented system by itself is statutory subject matter. Further, independent claim 1 is amended to recite *a memory operatively coupled* *to a processor that retains the method call interceptor,* which is clearly hardware and thus statutory. Furthermore, independent claim 14, as amended, recites a computer readable *storage* medium, which is statutory subject matter. By the standards set forth in the above decision, a computer implemented system in the form of software, hardware, or the combination of both clearly falls within the categories of statutory subject matter.

Additionally, the subject claims clearly produce a useful, concrete and tangible result.

> Because the claimed process applies the Boolean principle [abstract idea] ***to produce a useful, concrete, tangible result*** ... on its face the claimed process comfortably falls within the scope of §101. *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed.Cir. 1999) (Emphasis added); *See State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1601 (Fed.Cir.1998). The inquiry into patentability requires an examination of the contested claims to see if the claimed subject matter, as a whole, is a disembodied mathematical concept representing nothing more than a "law of nature" or an "abstract idea," or if the mathematical concept has been ***reduced to some practical application rendering it "useful."*** *AT&T* at 1357 *citing In re Alappat*, 33 F.3d 1526, 31 1544, 31 U.S.P.Q.2D (BNA) 1545, 1557 (Fed. Cir. 1994) (Emphasis added) (holding that more than an abstract idea was claimed because the claimed invention as a whole was directed toward forming a specific machine that produced the useful, concrete, and tangible result of a smooth waveform display).

Independent claims 1, 14 and 27 recite similar aspects, namely performing proxy pre-processing that includes machine learning to optimize remote method call invocation, which is a concrete, useful, and tangible result.

In view of the above, it is readily apparent that the claimed invention as recited in independent claims 1, 14, and 27 (and associated dependent claims 2-8 and 10-13) reduces to a practical application that produces a useful, concrete, tangible result; therefore, pursuant to *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358 (Fed.Cir. 1999), the subject claims are directed to statutory subject matter pursuant to 35 U.S.C. §101. Accordingly, this rejection should be withdrawn.

II. Rejection of Claims 1, 14, 15, 23 and 27 Under 35 U.S.C §112

Claims 1, 14, 15, 23 and 27 stand rejected under 35 U.S.C §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. This rejection should be withdrawn for at least the following reasons. Claims 1 and 27 have been amended to recite at least one of adapt or extend functionalities *of the system* (*See* page 2, lines 14-18). Similarly, claims 14, 15, and 23 are amended to recite at least one of adapt or extend functionalities *of the application code generic proxy* (*See* page 20, lines 13-14).

Claim 27 stands rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. 35 U.S.C. §112, sixth paragraph states.

> An element in a claim for a combination ***may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof***, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. (emphasis added)

Since the above-mentioned statute states that there does not need to be a recital of structure when using means-plus-function limitations, the claims do not need to specify a component that stores as stated by the Examiner on page 4 of the non-final Office Action (dated February 1, 2008). During the interview with the Examiner, it was requested the support in the specification for elements in claim 27 be presented.

The proxy 220 includes a customized proxy component 230 that can be written in application code by application programmers and which has access to the interception and routing functionality provided by system code. (*See* page11, lines 9-11.) Thus, the customized proxy component 230 operates as a "means for creating a base class proxy object from application code". Further, the customized proxy component 230 operates as a "means for creating the application code generic proxy".

The system includes a method call interceptor that intercepts a method call to an object and routes the method call to a proxy. (*See* page 3, lines 11-12.) The method call interceptor operates as a "means for intercepting a method call and for transferring control to the overridden base class method in the application code generic proxy". Additionally, *the system includes a method call intercepting component that intercepts a method call to an object and routes the method call to a proxy. The method call intercepting component is accessible to application code* (*See* pages 3-4, lines 30-1.) and operates as a "means for the interception to be made accessible to application code to at least one of adapt or extend functionalities of the system".

Access to the intercepting infrastructure 720 is facilitated by the presence of a message object 730 that provides a "means for accessing method call interception functionality". Further, *the method call information facilitates performing proxy pre-processing by providing the customized proxy 750 with information concerning the method call made by the method caller 710 (e.g., input parameters) and similarly facilitates proxy post-processing (e.g., by providing return parameter information).* Thus, the customized proxy 750 can function as a "means for retrieving information associated with a method call that can be intercepted by the interception functionality".

At page 11, lines 13-15, the subject specification indicates *the customized proxy component 230 can be, therefore, operable to adapt and/or extend the functionality provided by a conventional proxy (e.g., proxy 120, Prior Art Fig. 1), which facilitates mitigating flexibility problems associated with conventional systems.* Thus, the customized proxy component 230 can function as a "means for at least one of adapting or extending the functionality of object systems".

At page 30, lines 14-16, the subject specification indicates, *the system 1300 includes an intercepting infrastructure 1320 that can intercept such method calls and route the intercepted method calls to the proxies 1350.* Thus, the intercepting infrastructure 1320 can operate as a "means for the application code generic proxy to receive the intercepted method call". Additionally, at page 13, lines 23-26 *indicate the system 200 provides application programmers with a message object 270, access to proxies and the opportunity to perform proxy pre-processing and proxy post-processing, which facilitates adapting and extending the capabilities of conventional systems.* The message object 270 can operate as a "means for modifying the application code generic proxy operability based at least in part on the intercepted method call and externally supplied application code". Further, the Abstract indicates *the system also includes an application code generic proxy that after overriding a base class invocation method can receive intercepted method calls, can invoke methods on the object that the proxy images,* *can receive results from the object that the proxy images and can pass results to the entity that generated the intercepted method call.* The application code generic proxy can function as a "means for providing the overridden base class method with a call data structure associated with the intercepted method call". Furthermore, the customized proxy component 230 can operate as a "means for the application code generic proxy to invoke the method on the object" as indicated on page 11, lines 26-27.

At page 12, the subject specification indicates, *proxy pre-processing performed by the customized proxy 230 may have determined that a second remote object may be better suited to handle the method invoked by the method caller 210 than a first remote object.* Thus, the customized proxy 230 implements a "means for the application code generic proxy to perform proxy pre-processing that includes load-balancing, transaction processing and machine learning to optimize remote method call invocation prior to utilizing the means for the application code generic proxy to invoke the method on the object, wherein the optimization of the remote method call includes at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked".

The remoting infrastructure 1330 can operate as a "means for the application code generic proxy to receive a first result from the object" as seen at page 30, lines 18-20 of the subject specification, while the remoting infrastructure 620 can function as a "means for the application code generic proxy to return a second result to the entity that generated the intercepted method call" as seen at page 19, lines 2-5.

Moreover, the claims are considered part of the disclosure and thus supported. It is to be appreciated that the disclosed support is used for explanatory purposes to show support and is not to be construed as structural limitations. Accordingly, this rejection should be withdrawn and the claims allowed.

III. <u>Rejection of Claims 1-8 and 9-14 Under 35 U.S.C. §103(a)</u>

Claims 1-8 and 9-14 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Arnold, *et al.* (U.S. Patent No. 6,393,497), in view of Hollander, *et al.* (U.S. Patent No. 6,823,460), in further view of Clarke, *et al.* (U.S. Patent Publication No. 2002/0035642). It is respectfully submitted that this rejection is improper for at least the following reasons. Arnold, *et al.* alone or in combination with Hollander, *et al.* and/or Clarke, *et al.* does not teach or suggest each and every aspect of the subject claims.

> ***[T]he prior art reference (or references when combined) must teach or suggest all the claim limitations.*** *See* MPEP § 706.02(j). *See also KSR Int'l Co. v. Teleflex, Inc.*, 550 U. S. ____, 04-1350, slip op. at 14 (2007). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (emphasis added).

Applicant's subject claims relate to a system that employs application developer coded extensible proxies that have access to method interception and remote functionality and data. In particular, independent claims 1 and 14 relate to similar aspects, namely, ***optimizing the remote method call includes at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked.*** The cited references, alone or in combination, fail to teach this novel aspect.

Arnold, *et al.* relates to a system and method that employs a smart proxy as a wrapper around a stub in a distributed system. The system transmits a request for an object and receives a response to the request, which includes code that is employed to construct a representation of the requested object. The system creates the representation and employs it for processing calls to the object, local to the requesting object using the representation. However, Arnold, *et al.* fails to disclose a component for method call interception that can be accessed by application code to adapt and/or extend system functionalities of the system. Further, Arnold, *et al.* does not teach or suggest optimizing remote method calls by determining an amount of data sent to the object, determining a type of data sent to the object and/or determining which objects are invoked.

Hollander, *et al.* relates to a method of intercepting application program interface wherein an API interception control server in conjunction with a system call interception module loads into all active process spaces an API interception module. Further, the disclosed system executes user supplied code for determining return values of the API routine and to manipulate process level flow control structure to enable return control to user application. Additionally, an apparatus is provided to provide system call interception that includes means to obtain a list if active processes within the computer system, means to open processes within a computer system and means to issue notification messages associated with the system calls. However, the system fails to disclose an application code generic proxy that performs proxy pre-processing comprising at least machine learning to optimize remote method call invocation prior to invoking a method on a object, *wherein the optimization of the remote method call includes at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked.* Furthermore, Clarke, *et al.* fails to make up for the aforementioned deficiencies of Hollander, *et al.* and Arnold, *et al.*

Clarke, *et al.* simply teaches a system and method that manages network traffic by employing an intermediary node, such as a proxy, that implements a flow control algorithm to avoid network congestion. The system includes a server that returns an error response when it receives a request that it cannot handle from the client. This response is passed back to the client *via* a proxy, which recognizes the response type and learns a back off time for the server. Incase a disparate client sends a request to the server, the proxy returns a back off signal to the disparate client and reduces the number of requests reaching the congested server. However, Clarke, *et al.* is silent with respect to *at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked for optimizing a remote method call.*

Applicants' subject claims, in contrast, relate to a system that facilitates application developers creating proxies, accessing method call interception functionality, retrieving information associated with a method call that can be intercepted by the interception functionality and adapting and/or extending the functionality of object systems. The disclosed system facilitates method call interception wherein control passes from the method caller to a proxy, rather than from the method caller to a remote object. While such interception and routing is conventionally performed by system code, *the disclosed system facilitates gaining access to such interception and routing functionality.* Thus, the proxy includes a customized proxy component that can be written in application code by application programmers and which has access to the interception and routing functionality provided by system code (*e.g.,* operating system, intercepting infrastructure, remoting infrastructure, distributed object system). The customized proxy component can be, therefore, operable to adapt and/or extend the functionality provided by a conventional proxy. Furthermore, application code may be employed in actions including, but not limited to, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation. (*See* page 11, lines 3-25.) By providing application programmers with the opportunity to produce and custom craft the customized proxy and thus to engage in proxy pre-processing, the proxy can be adapted and/or extended beyond the capabilities provided by conventional systems. Since the customized proxy had the opportunity to perform proxy pre-processing, the invocation includes optimization of the amount of data sent to the remote object, the type of data sent to the remote object and which remote object(s), if any, are invoked. By way of illustration, the proxy pre-processing may have determined that a database value had previously been cached in the proxy, and thus the database value could be passed to the remote object and thus the database lookup may not occur. For example, in the web browser/stock market data feed example, a user may have previously inquired about a stock price for a company named AABBCC company. Before the stock price could be retrieved, the remote object may have had to look up a ticker symbol for the company (*e.g.,* AACo), and that ticker symbol may have been cached in the customized proxy. Thus, the ticker symbol, rather than the name of the company, may be sent to the remote object which can eliminate a duplicate lookup in the remote object. By way of further illustration, proxy pre-processing performed by the customized proxy may have determined that a second remote object may be better suited to handle the method invoked by the method caller than a first remote object, and thus the customized proxy may invoke the method on the second remote object instead of the first remote object. For example, returning to the browser/stock market data feed example, the customized proxy may have determined that recent stock market inquiries on a first remote object have taken sixty seconds, while a response time of five seconds is desired. Thus, the customized proxy may acquire the stock market price from a different remote object and compare the average response times to determine how more optimal response times can be achieved. Thus, increases in performance may be achieved for the method caller. The cited references are silent with respect to these novel aspects.

In view of at least the foregoing, it is readily apparent that Arnold, *et al.*, Hollander, *et al.* and Clarke, *et al.*, alone or in combination, do not teach or suggest applicants' claimed subject matter as recited in independent claims 1 and 14 (and claims 2-8 and 9-13 which respectively depend there from) and thus fail to make obvious the subject claims. Accordingly, this rejection should be withdrawn.

IV. Rejection of Claims 15 and 20-22 Under 35 U.S.C. §103(a)

Claims 15 and 20-22 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Colyer (U.S. Patent No. 5,903,725), in view of Clarke, *et al.* (U.S. Patent Publication No. 2002/0035642), in further view of Hollander, *et al.* (U.S. Patent No. 6,823,460). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Colyer, Clarke, *et al.* and Hollander, *et al.*, alone or in combination, fail to teach or suggest each and every limitation of the subject claims.

Applicants' subject claims relate to application developer coded extensible proxies that have access to method interception and remoting functionality and data. In particular, Independent claim 15, as amended, recites ***the optimizing the remote method call includes at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked***. Additionally, independent claim 15 recites, ***performing custom, user-directed application processing to at least one of monitor or control the processing of a message between the application code generic proxy and the object***. The cited references, alone or in combination, fail to disclose these novel features.

Colyer teaches a system that creates recoverable proxies to protect a server against invalid usage of proxy objects. Specifically, the system transparently re-creates proxy objects in a client of a client-server distributed processing system. On malfunction of a server, and consequently invalidity of the proxy objects, a proxy register object causes all proxy objects to be refreshed. However, Colyer fails to disclose interception of a method call that is made accessible to a developer to at least one of adapt or extend functionalities of a proxy and does not teach or suggest determination of an amount of data sent to the object, a type of data sent to the object or determination which objects are invoked. Further, Colyer is silent with respect to monitoring or controlling the processing of a message between the application code generic proxy and the object.

Clarke, *et al.* relates to a system and method to control network traffic by employing an intermediary node, such as a proxy, that implements a flow control algorithm to avoid network congestion and as discussed above does not teach or suggest at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked for optimizing a remote method call. Furthermore, Hollander, *et al.* does not cure the aforementioned deficiency with respect to Clarke, *et al.*, as seen *supra*.

Thus, it is readily apparent that the cited references, alone or in combination, do not teach or suggest applicants' claimed subject matter as recited in independent claim 15 (and claims 20-22 which respectively depend there from). Accordingly, withdrawal of this rejection is requested.

V. Rejection of Claims 17-19, 23 and 27 Under 35 U.S.C. §103(a)

Claims 17-19, 23 and 27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Colyer (U.S. Patent No. 5,903,725), Clarke, *et al.* (U.S. Patent Publication No. 2002/0035642), and Hollander, *et al.* (U.S. Patent No. 6,823,460), in further view of Arnold, *et al.* (U.S. Patent No. 6,393,497). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. The cited references, alone or in combination, fail to teach or suggest each and every limitation of applicants' claimed invention.

Independent claims 15, 23 and 27 recite similar aspects of a method call interception being made accessible to a developer to adapt and/or extend functionalities of a proxy wherein ***the remote method call is optimized by at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked***. As discussed above, Colyer and Clarke, *et al.*, alone or in combination do not disclose a system or method that intercepts a method call and optimizes the method call by determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked. Furthermore, Arnold, *et al.* and Hollander, *et al.* are silent with respect to a method call interceptor that is accessible to application code for adapting and/or extending system functionalities of a proxy and fail to remedy the aforementioned deficiencies of Colyer and Clarke, *et al.* Therefore, it is respectfully requested that this rejection be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP243US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,
AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731